Analog ist

if b then c_1 else c_2 fi $:=$ (assume (b); c_1) + (assume $(\neg b)$; c_2).

$$\text{(SKIP)}\ \frac{\text{stable}(P,R)}{\text{skip}:(P,R,G,P)} \qquad \frac{\text{stable}(P,R) \quad c:(P,R,G,P)}{c^*:(P,R,G,P)}\ \text{(LOOP)}$$

$$\text{(SEQ)}\ \frac{c_1:(P,R,G,P') \quad c_2:(P',R,G,Q)}{c_1;c_2:(P,R,G,Q)} \qquad \frac{c_1:(P,R,G,Q) \quad c_2:(P,R,G,Q)}{c_1+c_2:(P,R,G,Q)}\ \text{(CHOICE)}$$

Für primitive Befehle greifen wir zurück auf die unterliegende Separation-Logik:

Beachte, dass der Shared-State nicht geändert werden darf!

$$\text{(COM)}\ \frac{\vdash_{SL} \{A\}\ \text{com}\ \{B\} \qquad \text{modifies}(\text{com}) \cap \text{free}(R,G) = \emptyset}{\text{com}:(A,R,G,B).}$$

$\text{free}(\overline{x}.\Pi * P) := \text{free}(\Pi,P) \setminus \overline{x}$

Die Regel für die Parallelkomposition ist wie bei Rely-Guarantee, nur dass wir jetzt * nehmen:

$$\text{(PAR)}\ \frac{\begin{array}{l} c_1:(P_1, R \cup G_2, G_1, Q_1) \quad \text{modifies}(c_1) \cap \text{free}(c_2,P_2,Q_2)=\emptyset \\ c_2:(P_2, R \cup G_1, G_2, Q_2) \quad \text{modifies}(c_2) \cap \text{free}(c_1,P_1,Q_1)=\emptyset \end{array}}{c_1 \| c_2 : (P_1 * P_2, R, G_1 \cup G_2, Q_1 * Q_2).}$$

Die schwierigste Regel ist jetzt atomic c.

Zwei Schritte.

Prüfe zunächst, ob der atomare Block seine Spezifikation in leerer Umgebung erfüllt. (nicht einfach).

Prüfe dann, dass Pre- und Postcondition stabil gegenüber Rely.